NASA-TM-108578

IN-61
136108
P.35

SOFTWARE MANAGEMENT ENVIRONMENT (SME)

INSTALLATION GUIDE

JANUARY 1992

NASA

(NASA-TM-108578)   SOFTWARE
MANAGEMENT ENVIRONMENT (SME)
INSTALLATION GUIDE   (NASA)   35 p

N93-15578

Unclas
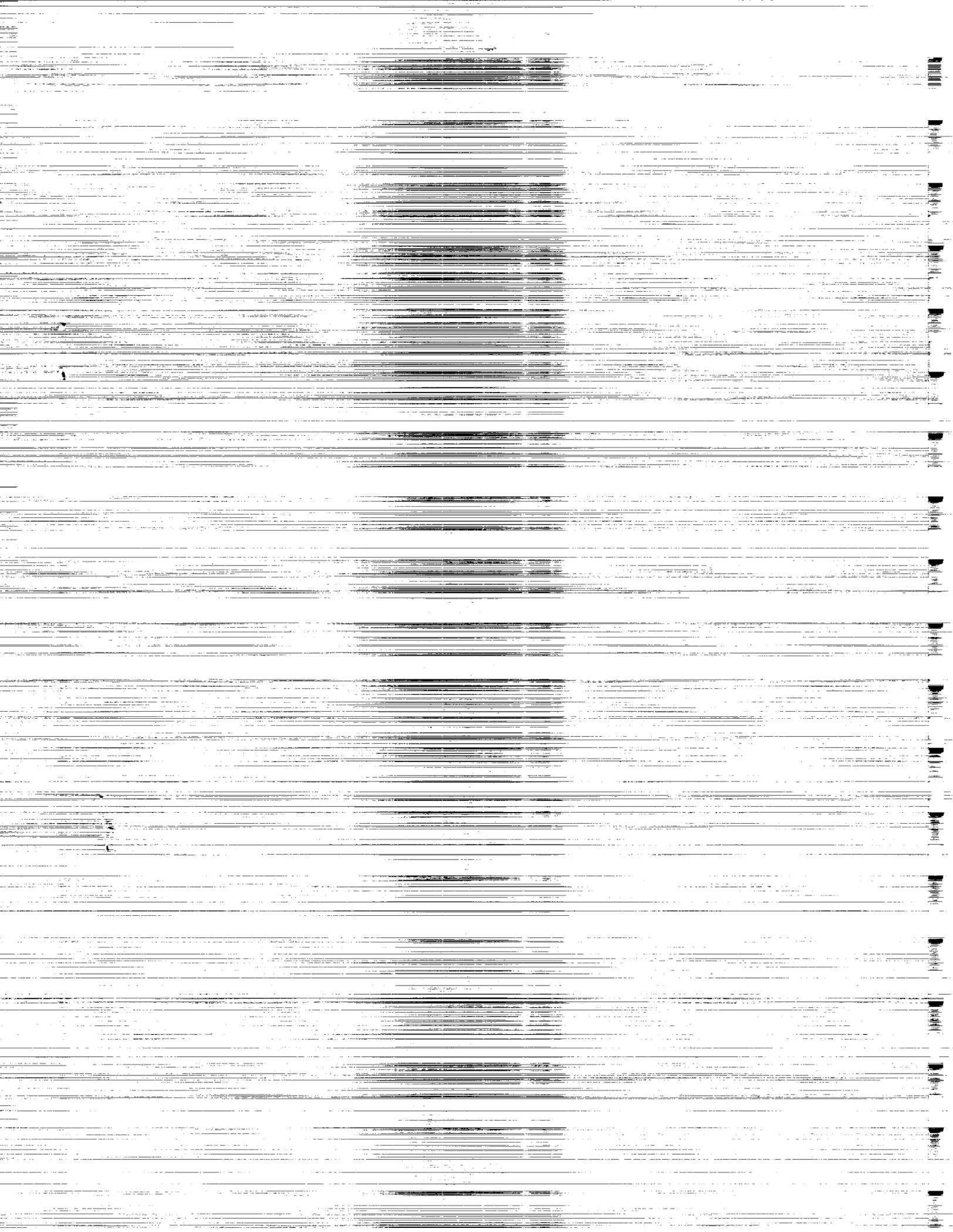
G3/61   0136108

SOFTWARE MANAGEMENT ENVIRONMENT (SME)

INSTALLATION GUIDE

JANUARY 1992

NASA

National Aeronautics and
Space Administration

Goddard Space Flight Center
Greenbelt, Maryland 20771

FOREWORD

The Software Engineering Laboratory (SEL) is an organization sponsored by the National Aeronautics and Space Administration/Goddard Space Flight Center (NASA/GSFC) and created to investigate the effectiveness of software engineering technologies when applied to the development of applications software. The SEL was created in 1976 and has three primary organizational members:

NASA/GSFC, Systems Development Branch

University of Maryland, Department of Computer Science

Computer Sciences Corporation, Systems Development Operation

The goals of the SEL are (1) to understand the software development process in the GSFC environment; (2) to measure the effect of various methodologies, tools, and models on this process; and (3) to identify and then to apply successful development practices. The activities, findings, and recommendations of the SEL are recorded in the Software Engineering Laboratory Series, a continuing series of reports that includes this document.

The major contributor to this document is

David Kistler (Computer Sciences Corporation)

Additionally, the following person contributed significantly:

Kellyann Jeletic (GSFC)

Single copies of this document can be obtained by writing to

Systems Development Branch
Code 552
Goddard Space Flight Center
Greenbelt, Maryland 20771
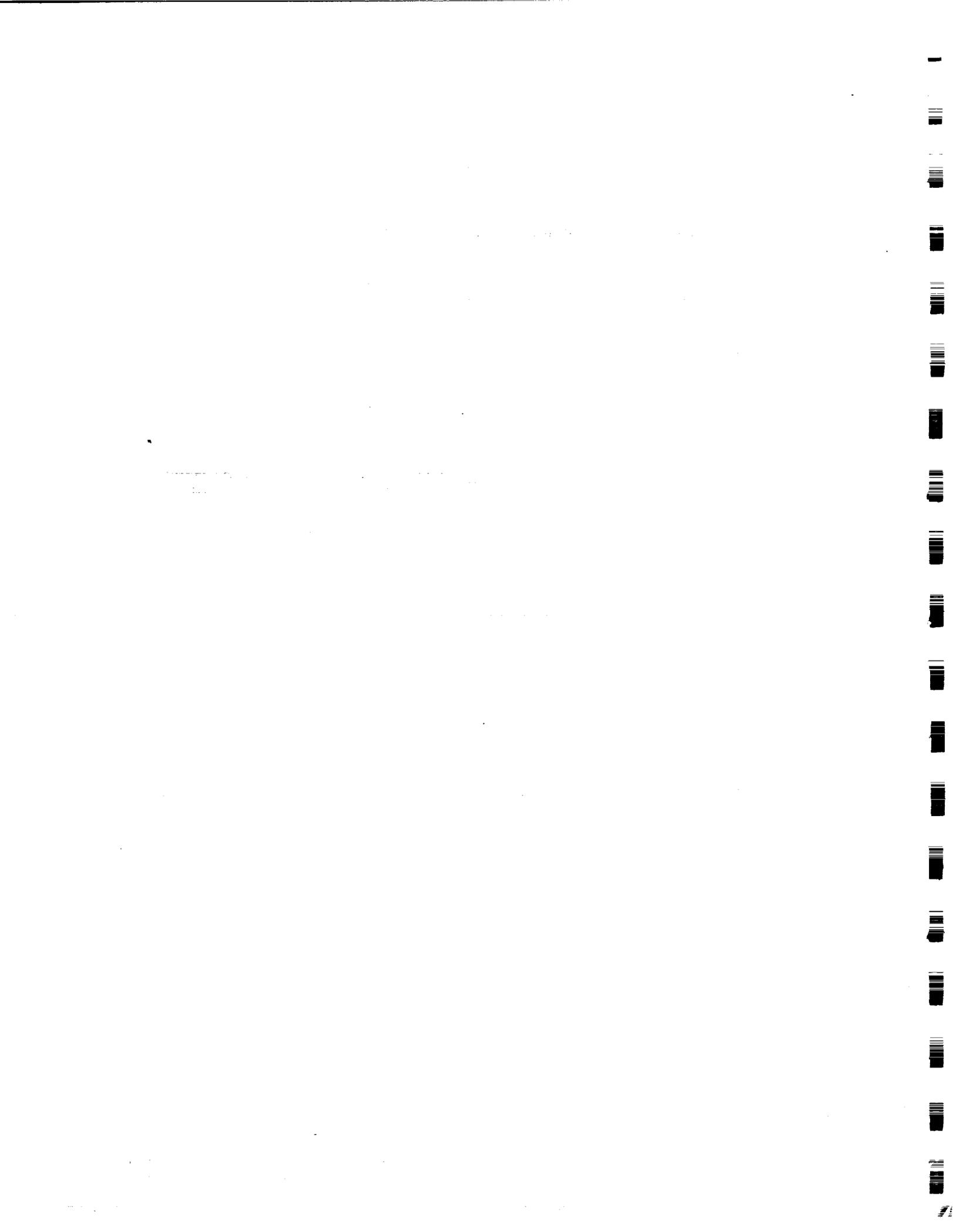
ABSTRACT

This document contains installation information for the Software Management Environment (SME), developed for the Systems Development Branch (Code 552) of the Flight Dynamics Division of Goddard Space Flight Center (GSFC). The SME provides an integrated set of management tools that can be used by software development managers in their day-to-day management and planning activities. This document provides a list of hardware and software requirements as well as detailed installation instructions and trouble-shooting information.

Table of Contents

Table of Contents (Cont'd)

viii
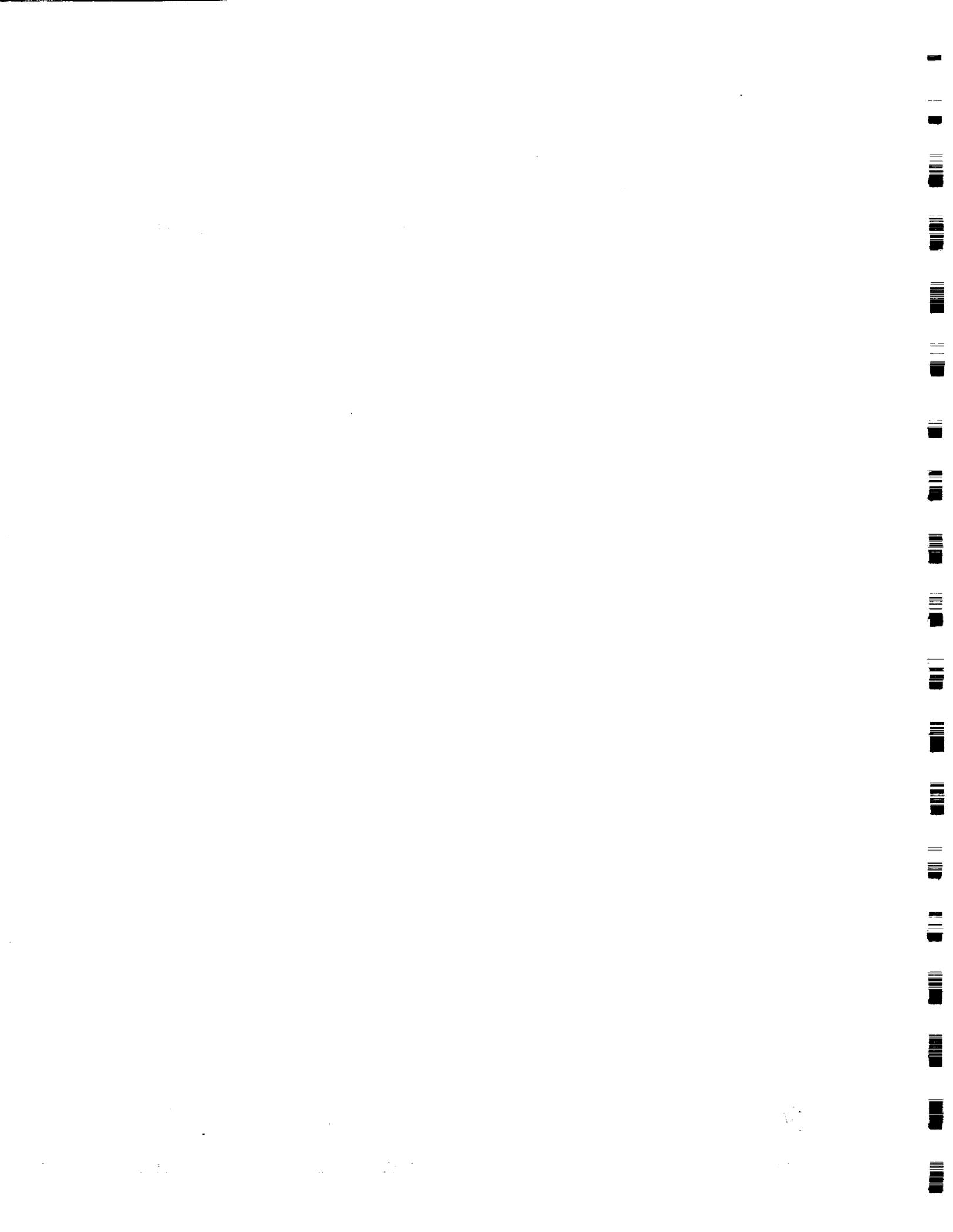
SECTION 1—DOCUMENT OVERVIEW

The Software Management Environment (SME) is being developed for the Systems Development Branch (Code 552) of the Flight Dynamics Division (FDD) of the Goddard Space Flight Center (GSFC). The SME provides an integrated set of management tools that can assist software development managers in their day-to-day management and planning activities.

This document contains the information you will need to install the SME software on a VAX computer connected to personal computer (PC) workstations. It is organized as follows:

- Section 2 presents an overview of the installation procedures and summarizes the minimum hardware, software, and communications requirements for the SME. General restrictions and assumptions are also summarized.

- Section 3 describes the SME directory structures as they exist on VAX computers and PC workstations, and presents a summary of the files and their contents.

- Section 4 describes how to install and configure the SME software.

- Section 5 provides troubleshooting hints for successful execution. General guidelines and specific anomalies are given.

- The appendix describes the SME configuration and support files required for execution.

10000272
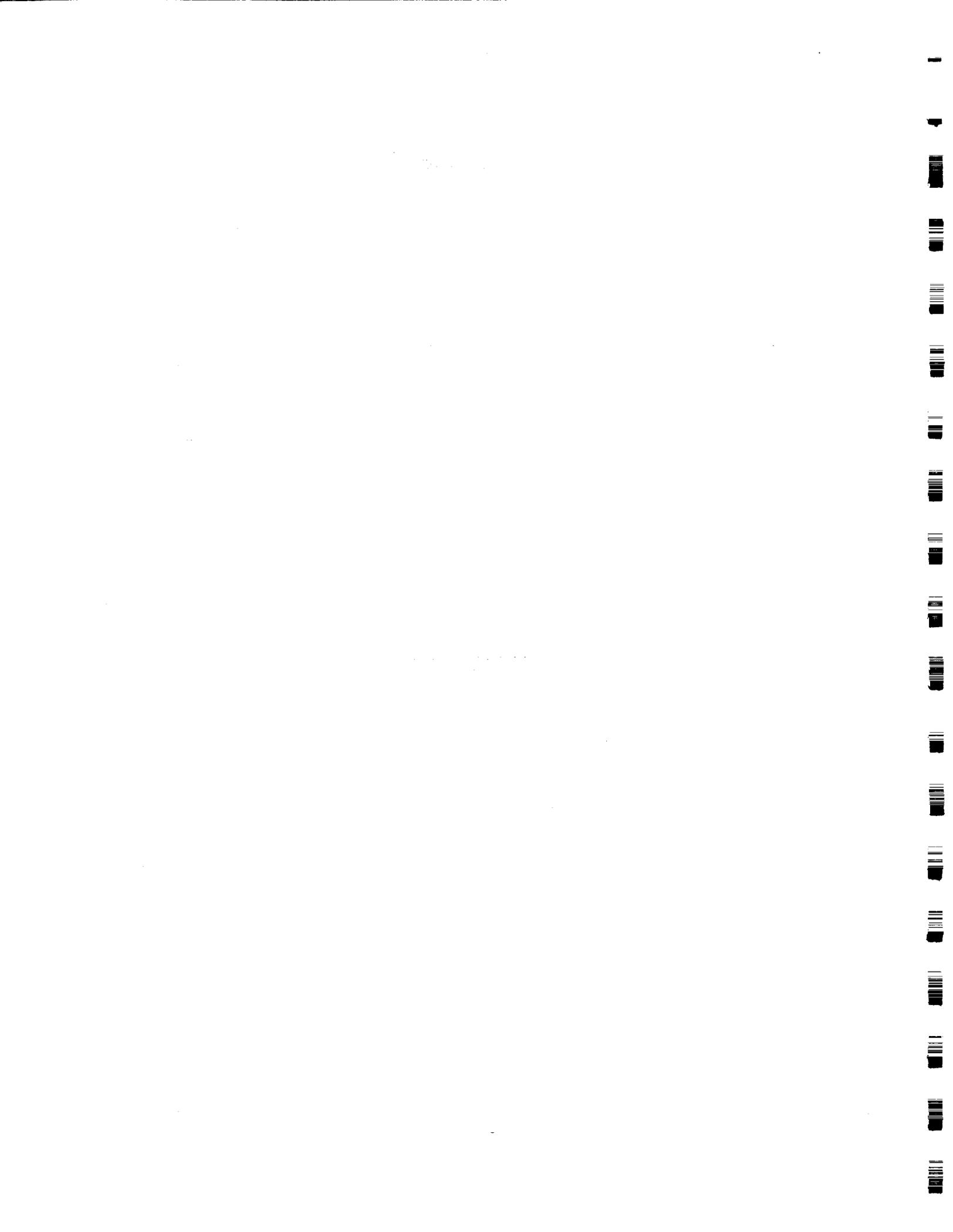
SECTION 2—INSTALLATION PROCEDURE OVERVIEW

The following sections describe the hardware, software, and communications configuration you will need to install and run the SME on VAX computers and PC workstations. General assumptions and restrictions are also presented.

2.1 INSTALLATION OVERVIEW

Because of the distributed architecture of the SME, there are two stages to complete installation: loading the VAX-resident portion of the software, and loading the PC-resident portion of the software.

The VAX-resident portion of the software is distributed on a 6250 bits per inch (bpi) 9-track tape created using the virtual memory system (VMS) backup utility. This tape contains all files that you will need to execute the SME. After you unload the tape onto the target VAX computer, the files will appear in the directory structure that the SME expects (Section 3.1). Section 4.1 presents step-by-step procedures for installing the VAX-resident portion of the SME.

The PC-resident portion of the SME software is distributed on a 360-kilobyte (kB) installation diskette that was formatted under Microsoft Disk Operation System (MS-DOS) Version 3.3. Section 4.2 presents complete instructions for installing the PC-resident portion of the SME software.

2.2 HARDWARE REQUIREMENTS

The VAX hardware and memory you will need to install and run the SME are as follows:

- VAX computer
- Asynchronous communications port
- 800-kB memory (peak working set size of 1500 pages)
- 3 megabytes (MB) minimum disk storage (5 MB optimum)

The PC hardware and memory you will need to install and run the SME:

- IBM PC or compatible
- Enhanced graphics adapter (EGA) color monitor[1]
- Serial communication port (EIA RS-232C)

---

[1] A video graphics array (VGA) color monitor may also be used for EGA graphics in VGA medium-resolution mode.

2-1

- 640-kB memory
- 125-kB disk storage

2.3 SOFTWARE REQUIREMENTS

The following lists the VAX and PC software you will need to install and run the SME:

- VAX—VMS operating system (Version 5.3 or higher)[1]
- PC—MS-DOS (Version 3.0 or higher)

Additionally, VAX LISP must be available in order to run the SME's expert system.

2.4 COMMUNICATIONS REQUIREMENTS

The following delineates the communications configuration required by the SME:

- Support for serial asynchronous communications
- Conformance to the EIA RS-232C communications standard
- Interface via a hard-wired connection (i.e., data terminal equipment (DTE) to DTE in a "null modem configuration") or via a modem (i.e., DTE to data communications equipment (DCE) to DTE)

2.5 ASSUMPTIONS AND RESTRICTIONS

The SME program assumes the SME user has appropriate READ, WRITE, and EXECUTE privileges for the directories created by the VAX SME software installation procedure. Additionally, the user must have a disk quota for creating files on the disk where the SME was installed.

The only installation-specific restriction concerns VAX user IDs. The maximum user ID length the current version of the SME will accept is 10 characters.

---

[1] The SME requires runtime support for VAX system service routines and for VAX Pascal Version 4.0.

The following sections provide an overview of the VAX and PC directory structures expected by the SME software. The files contained in the directories are also briefly described.

3.1 SME VAX DIRECTORY STRUCTURE

All files that the SME requires at runtime are distributed logically, based on usage, in a hierarchy of directories. The files needed by the SME include executable images, project data, models, and various definitions files. Figure 3-1 shows the directory structure that will be created by unloading the VAX SME software (Section 4.1) onto the target computer.
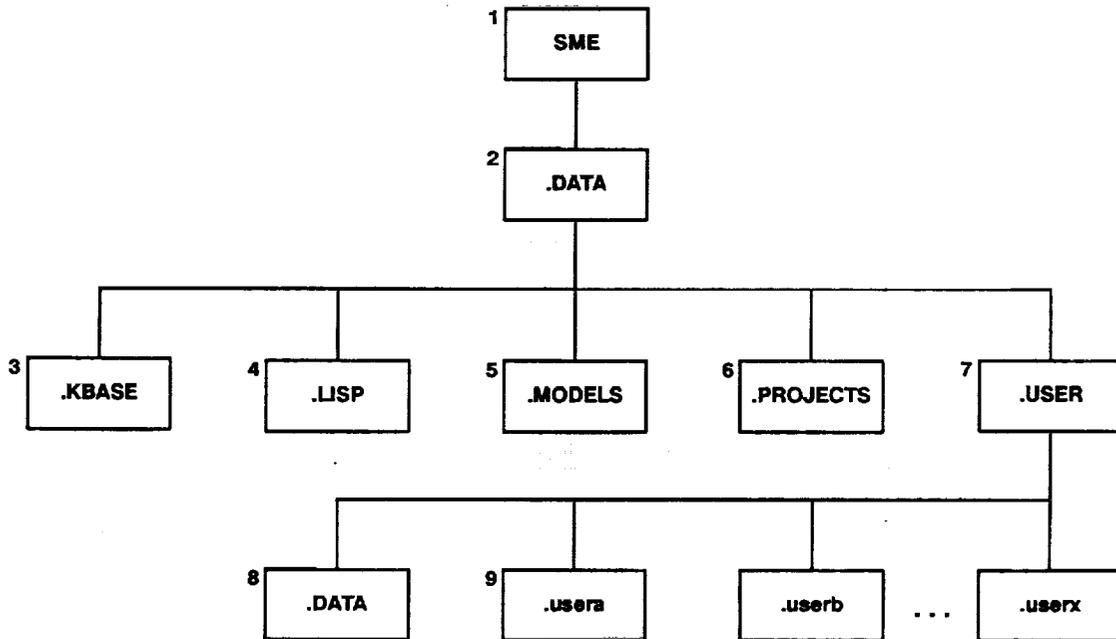
Figure 3-1. SME Directory Structure

The following is a list of the directories and a brief description of their contents. The numbers correspond to the numbers shown in Figure 3-1.

1. [SME] This directory contains the command files and executable images necessary to configure and run the SME software on the VAX. The directory also contains the [SME.DATA] directory. After execution of the SME, the [SME] directory will contain debug and user context files. As a result, **users will require write privileges for this directory.**

2. [SME.DATA] This directory contains the [KBASE], [LISP], [MODELS], [PROJECTS], and [USER] directories.

3. [SME.DATA.KBASE] This directory contains all knowledge base files that define explanations, factors, and reasons used by the trend assessment function under the SME growth monitor.

4. [SME.DATA.LISP] This directory contains all files required by the software to execute the VAX LISP expert system function under the SME rate monitor.

5. [SME.DATA.MODELS] This directory contains all models used by the SME at run time. The directory also contains files used to define the growth and profile measures used by the SME.

6. [SME.DATA.PROJECTS] This directory contains all the sample project data files provided on the installation tape.

7. [SME.DATA.USER] This directory contains the [SME.DATA.USER. DATA] directory and will contain all individual user directories created at runtime. **Users will require write privileges for this directory.**

8. [SME.DATA.USER.DATA] This directory contains all project-specific subjective data files created interactively by the user at runtime. These subjective data files are used with the knowledge base files for performing trend assessment under the SME growth monitor. **Users will require write privileges for this directory.**

9. [SME.DATA.USER.userx] This directory contains all project-specific phase estimate files created interactively by the user at runtime. The phase estimate files are used with the prediction function to record the basis of each prediction made by the user. **Users will require write privileges for this directory.**

3.2 SME VAX FILES

The SME requires several different types of files at program execution. The following sections describe these files. Additional information concerning the types of data used by the SME can be found in the *Software Management Environment (SME) Concepts and Architecture* (Reference 1).

3.2.1 Executable Images

The SME VAX software requires two executable images: SME_DRIVER.EXE and SME_EXEC.EXE. SME_DRIVER.EXE is simply a driver that enables the SME to execute either under a VAX VMS environment or under a VAX LISP environment. SME_EXEC.EXE functions as the executive-level driver for the system. Both executable images reside in the [SME] directory.

3-2

3.2.2 Compiled LISP Files

The SME VAX software requires several compiled VAX LISP files to run the expert system feature. These files reside in the [SME.DATA.LISP] directory. Note: the expert system is only available at those sites having VAX LISP available on the target VAX.

3.2.3 Model Files

The model files contain information used by the SME to create "guidelines" that, when compared to graphically plotted project data, can give the user an indication of how the project being plotted is behaving with respect to a "typical" project. The models are sequential text files read by the SME at runtime. All model files reside in [SME.DATA.MODELS] and have the file extension MDL.

3.2.4 Project Data Files

The sample project data files provided on the installation tape contain historical data used by the SME to generate plots that graphically represent a project's behavior over its development life cycle. The files are sequential text files that function as input to the SME and have the file extension DAT. All project data files reside in the directory [SME.DATA.PROJECTS]. A complete list of sample projects and available data for each project is provided with the distribution kit.

3.2.5 Definition Files

The SME reads a number of files at runtime to define key elements required by various SME functions. The files MEASURE.LST and PROFILE.LST are read at runtime to define the growth and profile measures used by the SME. These are sequential text files that reside in the directory [SME.DATA.MODELS]. Another file, DEFINITIONS.LST, resides in the [SME] directory and defines the functions used by the overall assessment feature.

Three files define the explanations, factors, and reasons, respectively, used by the knowledge base: EXPLAIN.KBS, FACTOR.KBS, and REASON.KBS. These are also sequential text files that reside in the directory [SME.DATA.KBASE].

One file, INTERPRETATIONS.DAT, resides in directory [SME.DATA.LISP] and contains text descriptions of mnemonic codes returned by the LISP expert system function. (This option is not available at sites that do not have VAX LISP.)

3.2.6 Command Files

There are three primary command files, SME.COM, SME_SETUP.COM, and SME_RUN.COM, all residing in the [SME] directory. SME_SETUP.COM defines a set of VAX logicals required by the SME at runtime, and SME_RUN.COM invokes the executable image. Both SME_SETUP.COM and SME_RUN.COM are invoked by SME.COM.

3.3 SME PC FILES

There are three EXE files in the SME directory on the PC: INSTALL.EXE, SME_CHAT.EXE, and PORTSTAT.EXE. INSTALL.EXE is the program provided on the diskette that performs the actual installation of the SME onto the hard drive of the target PC. SME_CHAT.EXE is the program that reads, interprets, and subsequently displays the information sent to the PC by the VAX. PORTSTAT.EXE is a utility program provided to help identify which PC communications port is attached to the VAX communications line. The PORTSTAT program's use is further described in Section 4.2.

SMESTART.BAT is a batch file that establishes the PC graphics mode and invokes the PC software. This file is created by the INSTALL program, described in Section 4.2.

There are five additional support files used by the SME PC software at runtime: EGAVGA.BGI (a graphic device driver); and GOTH.CHR, LITT.CHR, SANS.CHR, and TRIP.CHR (graphic fonts).

SECTION 4—SME INSTALLATION AND CONFIGURATION PROCEDURES

The following sections describe the procedures used to install the SME software on target VAX and PC computers.

4.1 VAX INSTALLATION PROCEDURE

After mounting the SME installation tape on a 9-track 6250-bpi density tape drive, issue the following command (where MUA0 is the local tape drive name):

allocate MUA0:

VMS will confirm the successful allocation with a message similar to the following:

%DCL−I−ALLOC, _1MUA0: allocated

Next, issue the following command:

mount/foreign/density=6250 MUA0:

VMS will respond with a confirmation message similar to the following:

%MOUNT−I−MOUNTED, INSTAL mounted on _1MUA0: (STLH1)

Next, unload the tape using the VMS backup utility by issuing the following command:

backup MUA0:install.sme/save_set/select=[sme.test...] − /noassist [target directory...]

The "target directory" is the directory into which you are unloading the SME installation tape. Be sure this directory specification includes the three trailing dots.

Finally, issue the following two commands to dismount and deallocate the tape:

dismount MUA0:

deallocate MUA0:

After installing the software from the tape, you can verify that all the files were unloaded successfully by referring to the listing of the tape creation log provided with the distribution kit.

4.2 SME PC INSTALLATION PROCEDURE

Insert the SME installation diskette into one of the PC disk drives. Change the default drive to reference the drive containing the diskette. For example, if you inserted the diskette in the B drive, enter the following at the DOS prompt:

b:

4-1

Before proceeding, you must know two facts about the particular PC on which you are installing the SME:

1. The number of the PC communications port attached to the VAX communications line (normally 1 or 2)

2. The baud rate of the communications line to the VAX (e.g., 4800 or 9600)

The definitive source for this information is the PC system coordinator for your organization. You can also use the SME PORTSTAT utility, included on the installation diskette, to assist in determining the appropriate number to reference the PC communications port. To run this utility, enter:

portstat

For each PC communications port found, the utility generates one of the two following messages:

*   comX: status is HHHH

*   comX: is not present

In these messages, X will be the port's number (i.e., 1 or 2), and HHHH will be the port's status displayed in hexadecimal.

If one communications port is reported as not present, you may assume the other port is attached to the VAX communications line. Some PCs, however, have multiple communications ports, and a status code will be reported for each port. In this case, consult the PC system coordinator to determine the proper port to reference.

Once you have determined the port number and baud rate for the VAX communications line, you can complete the installation by entering:

install

The INSTALL program will prompt you for the port number and baud rate before installing the needed SME PC software in the directory C:\SME on the hard disk.

4.3   VAX COMMAND FILE CONFIGURATION

Before you can execute the SME software successfully, you must resolve a few administrative and configuration issues.

First, you must provide all prospective SME users with VAX logon IDs, disk space allocation quotas, access to all SME files, and write privileges to the [SME], [SME.DATA.USER], and [SME.DATA.USER.DATA] directories.

Next, you must modify the command file SME_SETUP.COM that resides in the [SME] directory to reference the appropriate disk name correctly. Edit the file and change all

4-2

occurrences of "STL_DISK1" to the name of the disk on which you placed the SME files.

Finally, you must provide each user with a LOGIN.COM file that defines a logical "SME$SYSTEM" referencing your disk name concatenated with the [SME] directory. The LOGIN.COM should also define a VAX symbol "SME" that invokes the SME.COM file in SME$SYSTEM, so that the user can invoke the VAX portion of the software by entering:

SME

Then, if VAX LISP is available, the user will be able to invoke the SME with the expert system by entering:

SME EXPERT

Complete instructions for running the SME are provided in the *Software Management Environment (SME) User Reference Material* (Reference 1). A sample LOGIN.COM file is provided on the installation tape and resides in the [SME] directory as LOGIN_USER.COM.

4-3

SECTION 5—TROUBLESHOOTING HINTS

The following sections provide some procedures to follow if you encounter problems when running the SME software.

5.1 GENERAL GUIDELINES

Once you have successfully installed the SME and tested all the options, you should create a backup tape. In the event that something catastrophic occurs and you have to reinstall the software, you will be installing your own configured version.

SME files can be edited, but with the exception of command (.COM) files, they should not be. Invalid characters, delimiters, or formats could inadvertently be introduced that would make further troubleshooting difficult if not impossible. If the files become hopelessly corrupted, you will probably have to reinstall the software.

If you are a first-time SME user, upon initialization you will receive the message that your context file could not be read and that default values are being used. This is not an error. The SME software creates a context file for you in the [SME] directory with a file name CONTEXT_userid.DAT. Assuming your context file does not get deleted, subsequent executions of the SME will not generate this message. Depending on the amount of use the SME receives, periodic purging of user context files is advised.

5.2 SPECIFIC ANOMALIES

The following list describes potential problems and corrective actions.

1. Problem: SME fails to execute.

    Response: Ensure that you are in the [SME] root directory and that SME_DRIVER.EXE and SME_EXEC.EXE are present.

                If you entered the "expert" parameter when attempting to run the software, ensure that LISP is present. If LISP is not present, reexecute the SME without the "expert" parameter.

2. Problem: SME fails to initialize properly (executive menu never appears).

    Response: Refer to the table in the appendix to check the logical references in SME_SETUP.COM; reconfigure, if necessary.

                Ensure MEASURE.LST and PROFILE.LST files are in the directory referenced by the "USRDATA" logical.

3. Problem: SME appears to "hang" when entering the PREDICT option of the growth monitor.

    Response: Ensure that your userid has write privileges for [SME.DATA.USER.userid] directory.

                Ensure that your userid has a quota for creating files and that the quota has not been exceeded.

                Ensure that the logical "usrdata" in SME_SETUP.COM is not concatenated with a second logical containing a trailing dot.

4. Problem: SME appears to "hang" when attempting to change a reason in the ASSESS option of the growth monitor.

    Response: See response 3.

5. Problem: SME appears to "hang" when entering the OVERALL ASSESSMENT option of the executive menu.

    Response: Ensure that the DEFINITIONS.LST file is in the [SME] root directory.

6. Problem: SME generates "Unable to write context file" message upon exit.

    Response: Ensure that your userid has a quota for creating files and that the quota has not been exceeded.

7. Problem: SME appears to "hang" when generating a graph.

    Response: Suspect corrupted data. See if any model or data file for the project and/or measure has a different version number (higher) or create date (later) than the other model or data files. If this is the case and the original file still exists, delete the newer file. If the original file does not exist (it may have been purged), restore the original file from a backup tape. It might be possible to reconstruct the file by hand, but it is highly unlikely.

APPENDIX—SUPPORT FILE REFERENCE LIST

The SME requires that five logical references be defined at runtime, which is performed by SME_SETUP.COM. The following listing presents the "logicals" and the directory referenced by each logical after installation.

Logical	Directory
PRJDATA	diskname:[SME.DATA.PROJECTS]
MDLDATA	diskname:[SME.DATA.MODELS]
KBSDATA	diskname:[SME.DATA.KBASE]
LSPDATA	diskname:[SME.DATA.LISP]
USRDATA	diskname:[SME.DATA.USER]

You may redefine the logicals to reference different directories by editing SME_SETUP.COM. The SME, however, expects the contents of the directories referenced by the logicals to be what is described in Section 3. For example, the models, MEASURE.LST, and PROFILE.LST files must reside in whatever directory is referenced by the logical MDLDATA.

The following listing presents what the SME expects in the directories referenced by each logical. It is a summary of the information contained in Section 3 and is included as a convenience.

Logical	Expected Contents
PRJDATA	sample project data supplied with the installation tape
MDLDATA	all models, MEASURE.LST, PROFILE.LST
KBSDATA	EXPLAIN.KBS, FACTOR.KBS, REASON.KBS
LSPDATA	compiled (FAS) LISP files, INTERPRETATIONS.DAT
USRDATA	DATA.DIR, user.DIR

Refer to Section 4.3 for more information on modifying the SME_SETUP.COM file.

GLOSSARY

bpi	bits per inch
DCE	data communications equipment
DTE	data terminal equipment
EGA	enhanced graphics adapter
FDD	Flight Dynamics Division
GSFC	Goddard Space Flight Center
kB	kilobyte
MB	megabyte
MS-DOS	Microsoft Disk Operating System
PC	personal computer
SME	Software Management Environment
VGA	video graphics array
VMS	virtual memory system
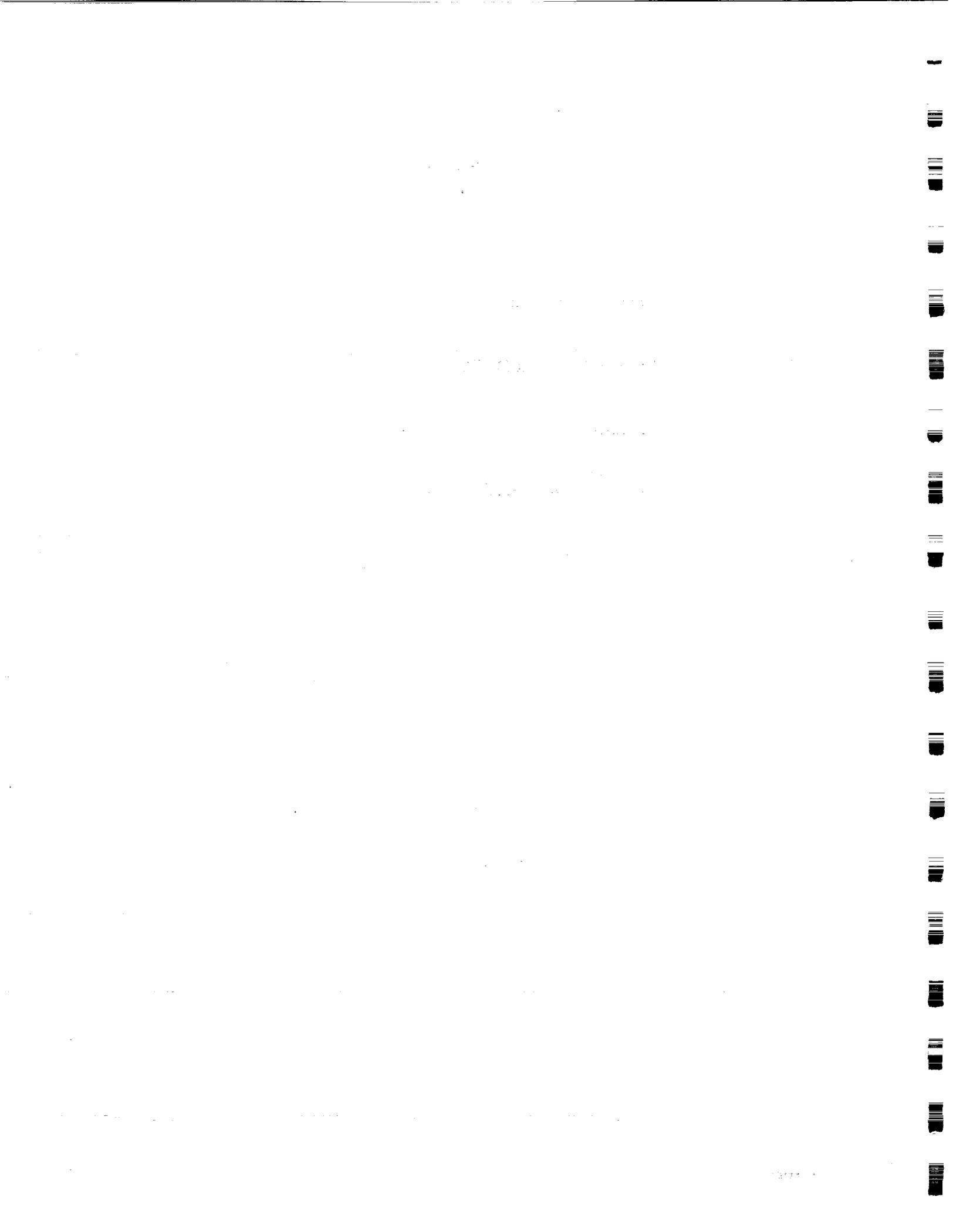
REFERENCES

1. Software Engineering Laboratory, SEL-89-003, *Software Management Environment (SME) Concepts and Architecture*, W. Decker and J. Valett, August 1989

2. Computer Sciences Corporation, 552-FDD-91/067, *Software Management Environment (SME) User Reference Material*, R. Hendrick and D. Kistler, September 1991

STANDARD BIBLIOGRAPHY OF SEL LITERATURE

The technical papers, memorandums, and documents listed in this bibliography are organized into two groups. The first group is composed of documents issued by the Software Engineering Laboratory (SEL) during its research and development activities. The second group includes materials that were published elsewhere but pertain to SEL activities.

SEL-ORIGINATED DOCUMENTS

SEL-76-001, *Proceedings From the First Summer Software Engineering Workshop*, August 1976

SEL-77-002, *Proceedings From the Second Summer Software Engineering Workshop*, September 1977

SEL-77-004, *A Demonstration of AXES for NAVPAK*, M. Hamilton and S. Zeldin, September 1977

SEL-77-005, *GSFC NAVPAK Design Specifications Languages Study*, P. A. Scheffer and C. E. Velez, October 1977

SEL-78-005, *Proceedings From the Third Summer Software Engineering Workshop*, September 1978

SEL-78-006, *GSFC Software Engineering Research Requirements Analysis Study*, P. A. Scheffer and C. E. Velez, November 1978

SEL-78-007, *Applicability of the Rayleigh Curve to the SEL Environment*, T. E. Mapp, December 1978

SEL-78-302, *FORTRAN Static Source Code Analyzer Program (SAP) User's Guide (Revision 3)*, W. J. Decker, W. A. Taylor, et al., July 1986

SEL-79-002, *The Software Engineering Laboratory: Relationship Equations*, K. Freburger and V. R. Basili, May 1979

SEL-79-003, *Common Software Module Repository (CSMR) System Description and User's Guide*, C. E. Goorevich, A. L. Green, and S. R. Waligora, August 1979

SEL-79-004, *Evaluation of the Caine, Farber, and Gordon Program Design Language (PDL) in the Goddard Space Flight Center (GSFC) Code 580 Software Design Environment*, C. E. Goorevich, A. L. Green, and W. J. Decker, September 1979

SEL-79-005, *Proceedings From the Fourth Summer Software Engineering Workshop*, November 1979

SEL-80-002, *Multi-Level Expression Design Language-Requirement Level (MEDL-R) System Evaluation*, W. J. Decker and C. E. Goorevich, May 1980

SEL-80-003, *Multimission Modular Spacecraft Ground Support Software System (MMS/ GSSS) State-of-the-Art Computer Systems/Compatibility Study*, T. Welden, M. McClellan, and P. Liebertz, May 1980

SEL-80-005, *A Study of the Musa Reliability Model*, A. M. Miller, November 1980

SEL-80-006, *Proceedings From the Fifth Annual Software Engineering Workshop*, November 1980

SEL-80-007, *An Appraisal of Selected Cost/Resource Estimation Models for Software Systems*, J. F. Cook and F. E. McGarry, December 1980

SEL-80-008, *Tutorial on Models and Metrics for Software Management and Engineering*, V. R. Basili, 1980

SEL-81-008, *Cost and Reliability Estimation Models (CAREM) User's Guide*, J. F. Cook and E. Edwards, February 1981

SEL-81-009, *Software Engineering Laboratory Programmer Workbench Phase 1 Evaluation*, W. J. Decker and F. E. McGarry, March 1981

SEL-81-011, *Evaluating Software Development by Analysis of Change Data*, D. M. Weiss, November 1981

SEL-81-012, *The Rayleigh Curve as a Model for Effort Distribution Over the Life of Medium Scale Software Systems*, G. O. Picasso, December 1981

SEL-81-013, *Proceedings of the Sixth Annual Software Engineering Workshop*, December 1981

SEL-81-014, *Automated Collection of Software Engineering Data in the Software Engineering Laboratory (SEL)*, A. L. Green, W. J. Decker, and F. E. McGarry, September 1981

SEL-81-101, *Guide to Data Collection*, V. E. Church, D. N. Card, F. E. McGarry, et al., August 1982

SEL-81-104, *The Software Engineering Laboratory*, D. N. Card, F. E. McGarry, G. Page, et al., February 1982

SEL-81-107, *Software Engineering Laboratory (SEL) Compendium of Tools (Revision 1)*, W. J. Decker, W. A. Taylor, E. J. Smith, et al., February 1982

SEL-81-110, *Evaluation of an Independent Verification and Validation (IV&V) Methodology for Flight Dynamics*, G. Page, F. E. McGarry, and D. N. Card, June 1985

SEL-81-205, *Recommended Approach to Software Development*, F. E. McGarry, G. Page, S. Eslinger, et al., April 1983

SEL-82-001, *Evaluation of Management Measures of Software Development*, G. Page, D. N. Card, and F. E. McGarry, September 1982, vols. 1 and 2

SEL-82-004, *Collected Software Engineering Papers: Volume 1*, July 1982

SEL-82-007, *Proceedings of the Seventh Annual Software Engineering Workshop*, December 1982

SEL-82-008, *Evaluating Software Development by Analysis of Changes: The Data From the Software Engineering Laboratory*, V. R. Basili and D. M. Weiss, December 1982

SEL-82-102, *FORTRAN Static Source Code Analyzer Program (SAP) System Description (Revision 1)*, W. A. Taylor and W. J. Decker, April 1985

SEL-82-105, *Glossary of Software Engineering Laboratory Terms*, T. A. Babst, M. G. Rohleder, and F. E. McGarry, October 1983

SEL-82-1006, *Annotated Bibliography of Software Engineering Laboratory Literature*, L. Morusiewicz and J. Valett, November 1991

SEL-83-001, *An Approach to Software Cost Estimation*, F. E. McGarry, G. Page, D. N. Card, et al., February 1984

SEL-83-002, *Measures and Metrics for Software Development*, D. N. Card, F. E. McGarry, G. Page, et al., March 1984

SEL-83-003, *Collected Software Engineering Papers: Volume II*, November 1983

SEL-83-006, *Monitoring Software Development Through Dynamic Variables*, C. W. Doerflinger, November 1983

SEL-83-007, *Proceedings of the Eighth Annual Software Engineering Workshop*, November 1983

SEL-83-106, *Monitoring Software Development Through Dynamic Variables (Revision 1)*, C. W. Doerflinger, November 1989

SEL-84-003, *Investigation of Specification Measures for the Software Engineering Laboratory (SEL)*, W. W. Agresti, V. E. Church, and F. E. McGarry, December 1984

SEL-84-004, *Proceedings of the Ninth Annual Software Engineering Workshop*, November 1984

SEL-84-101, *Manager's Handbook for Software Development (Revision 1)*, L. Landis, F. E. McGarry, S. Waligora, et al., November 1990

SEL-85-001, *A Comparison of Software Verification Techniques*, D. N. Card, R. W. Selby, Jr., F. E. McGarry, et al., April 1985

SEL-85-002, *Ada Training Evaluation and Recommendations From the Gamma Ray Observatory Ada Development Team*, R. Murphy and M. Stark, October 1985

SEL-85-003, *Collected Software Engineering Papers: Volume III*, November 1985

SEL-85-004, *Evaluations of Software Technologies: Testing, CLEANROOM, and Metrics*, R. W. Selby, Jr., and V. R. Basili, May 1985

SEL-85-005, *Software Verification and Testing*, D. N. Card, E. Edwards, F. McGarry, and C. Antle, December 1985

SEL-85-006, *Proceedings of the Tenth Annual Software Engineering Workshop*, December 1985

SEL-86-001, *Programmer's Handbook for Flight Dynamics Software Development*, R. Wood and E. Edwards, March 1986

SEL-86-002, *General Object-Oriented Software Development*, E. Seidewitz and M. Stark, August 1986

SEL-86-003, *Flight Dynamics System Software Development Environment (FDS/SDE) Tutorial*, J. Buell and P. Myers, July 1986

SEL-86-004, *Collected Software Engineering Papers: Volume IV*, November 1986

SEL-86-005, *Measuring Software Design*, D. N. Card et al., November 1986

SEL-86-006, *Proceedings of the Eleventh Annual Software Engineering Workshop*, December 1986

SEL-87-001, *Product Assurance Policies and Procedures for Flight Dynamics Software Development*, S. Perry et al., March 1987

SEL-87-002, *Ada® Style Guide (Version 1.1)*, E. Seidewitz et al., May 1987

SEL-87-003, *Guidelines for Applying the Composite Specification Model (CSM)*, W. W. Agresti, June 1987

SEL-87-004, *Assessing the Ada® Design Process and Its Implications: A Case Study*, S. Godfrey, C. Brophy, et al., July 1987

SEL-87-008, *Data Collection Procedures for the Rehosted SEL Database*, G. Heller, October 1987

SEL-87-009, *Collected Software Engineering Papers: Volume V*, November 1987

SEL-87-010, *Proceedings of the Twelfth Annual Software Engineering Workshop*, December 1987

SEL-88-001, *System Testing of a Production Ada Project: The GRODY Study*, J. Seigle, L. Esker, and Y. Shi, November 1988

SEL-88-002, *Collected Software Engineering Papers: Volume VI*, November 1988

SEL-88-003, *Evolution of Ada Technology in the Flight Dynamics Area: Design Phase Analysis*, K. Quimby and L. Esker, December 1988

SEL-88-004, *Proceedings of the Thirteenth Annual Software Engineering Workshop*, November 1988

SEL-88-005, *Proceedings of the First NASA Ada User's Symposium*, December 1988

SEL-89-002, *Implementation of a Production Ada Project: The GRODY Study*, S. Godfrey and C. Brophy, September 1989

SEL-89-003, *Software Management Environment (SME) Concepts and Architecture*, W. Decker and J. Valett, August 1989

SEL-89-004, *Evolution of Ada Technology in the Flight Dynamics Area: Implementation/ Testing Phase Analysis*, K. Quimby, L. Esker, L. Smith, M. Stark, and F. McGarry, November 1989

SEL-89-005, *Lessons Learned in the Transition to Ada From FORTRAN at NASA/ Goddard*, C. Brophy, November 1989

SEL-89-006, *Collected Software Engineering Papers: Volume VII*, November 1989

SEL-89-007, *Proceedings of the Fourteenth Annual Software Engineering Workshop*, November 1989

SEL-89-008, *Proceedings of the Second NASA Ada Users' Symposium*, November 1989

SEL-89-101, *Software Engineering Laboratory (SEL) Database Organization and User's Guide (Revision 1)*, M. So, G. Heller, S. Steinberg, K. Pumphrey, and D. Spiegel, February 1990

SEL-90-001, *Database Access Manager for the Software Engineering Laboratory (DAMSEL) User's Guide*, M. Buhler, K. Pumphrey, and D. Spiegel, March 1990

SEL-90-002, *The Cleanroom Case Study in the Software Engineering Laboratory: Project Description and Early Analysis*, S. Green et al., March 1990

SEL-90-003, *A Study of the Portability of an Ada System in the Software Engineering Laboratory (SEL)*, L. O. Jun and S. R. Valett, June 1990

SEL-90-004, *Gamma Ray Observatory Dynamics Simulator in Ada (GRODY) Experiment Summary*, T. McDermott and M. Stark, September 1990

SEL-90-005, *Collected Software Engineering Papers: Volume VIII*, November 1990

SEL-90-006, *Proceedings of the Fifteenth Annual Software Engineering Workshop*, November 1990

SEL-91-001, *Software Engineering Laboratory (SEL) Relationships, Models, and Management Rules*, W. Decker, R. Hendrick, and J. Valett, February 1991

SEL-91-003, *Software Engineering Laboratory (SEL) Ada Performance Study Report*, E. W. Booth and M. E. Stark, July 1991

SEL-91-004, *Software Engineering Laboratory (SEL) Cleanroom Process Model*, S. Green, November 1991

SEL-91-005, *Collected Software Engineering Papers: Volume IX*, November 1991

SEL-91-102, *Software Engineering Laboratory (SEL) Data and Information Policy (Revision 1)*, F. McGarry, August 1991

SEL-92-001, *Software Management Environment (SME) Installation Guide*, D. Kistler, January 1992

SEL-RELATED LITERATURE

[4]Agresti, W. W., V. E. Church, D. N. Card, and P. L. Lo, "Designing With Ada for Satellite Simulation: A Case Study," *Proceedings of the First International Symposium on Ada for the NASA Space Station*, June 1986

[2]Agresti, W. W., F. E. McGarry, D. N. Card, et al., "Measuring Software Technology," *Program Transformation and Programming Environments*. New York: Springer-Verlag, 1984

[1]Bailey, J. W., and V. R. Basili, "A Meta-Model for Software Development Resource Expenditures," *Proceedings of the Fifth International Conference on Software Engineering*. New York: IEEE Computer Society Press, 1981

[8]Bailey, J. W., and V. R. Basili, "Software Reclamation: Improving Post-Development Reusability," *Proceedings of the Eighth Annual National Conference on Ada Technology*, March 1990

[1]Basili, V. R., "Models and Metrics for Software Management and Engineering," *ASME Advances in Computer Technology*, January 1980, vol. 1

Basili, V. R., *Tutorial on Models and Metrics for Software Management and Engineering*. New York: IEEE Computer Society Press, 1980 (also designated SEL-80-008)

[3]Basili, V. R., "Quantitative Evaluation of Software Methodology," *Proceedings of the First Pan-Pacific Computer Conference*, September 1985

[7]Basili, V. R., *Maintenance = Reuse-Oriented Software Development*, University of Maryland, Technical Report TR-2244, May 1989

[7]Basili, V. R., *Software Development: A Paradigm for the Future*, University of Maryland, Technical Report TR-2263, June 1989

[8]Basili, V. R., "Viewing Maintenance of Reuse-Oriented Software Development," *IEEE Software*, January 1990

[1]Basili, V. R., and J. Beane, "Can the Parr Curve Help With Manpower Distribution and Resource Estimation Problems?," *Journal of Systems and Software*, February 1981, vol. 2, no. 1

[9]Basili, V. R., and G. Caldiera, *A Reference Architecture for the Component Factory*, University of Maryland, Technical Report TR-2607, March 1991

[1]Basili, V. R., and K. Freburger, "Programming Measurement and Estimation in the Software Engineering Laboratory," *Journal of Systems and Software*, February 1981, vol. 2, no. 1

[3]Basili, V. R., and N. M. Panlilio-Yap, "Finding Relationships Between Effort and Other Variables in the SEL," *Proceedings of the International Computer Software and Applications Conference*, October 1985

[4]Basili, V. R., and D. Patnaik, *A Study on Fault Prediction and Reliability Assessment in the SEL Environment*, University of Maryland, Technical Report TR-1699, August 1986

[2]Basili, V. R., and B. T. Perricone, "Software Errors and Complexity: An Empirical Investigation," *Communications of the ACM*, January 1984, vol. 27, no. 1

[1]Basili, V. R., and T. Phillips, "Evaluating and Comparing Software Metrics in the Software Engineering Laboratory," *Proceedings of the ACM SIGMETRICS Symposium/ Workshop: Quality Metrics*, March 1981

[3]Basili, V. R., and C. L. Ramsey, "ARROWSMITH-P—A Prototype Expert System for Software Engineering Management," *Proceedings of the IEEE/MITRE Expert Systems in Government Symposium*, October 1985

Basili, V. R., and J. Ramsey, *Structural Coverage of Functional Testing*, University of Maryland, Technical Report TR-1442, September 1984

Basili, V. R., and R. Reiter, "Evaluating Automatable Measures for Software Development," *Proceedings of the Workshop on Quantitative Software Models for Reliability, Complexity, and Cost*. New York: IEEE Computer Society Press, 1979

[5]Basili, V. R., and H. D. Rombach, "Tailoring the Software Process to Project Goals and Environments," *Proceedings of the 9th International Conference on Software Engineering*, March 1987

[5]Basili, V. R., and H. D. Rombach, "T A M E: Tailoring an Ada Measurement Environment," *Proceedings of the Joint Ada Conference*, March 1987

[5]Basili, V. R., and H. D. Rombach, "T A M E: Integrating Measurement Into Software Environments," University of Maryland, Technical Report TR-1764, June 1987

[6]Basili, V. R., and H. D. Rombach, "The TAME Project: Towards Improvement-Oriented Software Environments," *IEEE Transactions on Software Engineering*, June 1988

[7]Basili, V. R., and H. D. Rombach, *Towards A Comprehensive Framework for Reuse: A Reuse-Enabling Software Evolution Environment*, University of Maryland, Technical Report TR-2158, December 1988

[8]Basili, V. R., and H. D. Rombach, *Towards A Comprehensive Framework for Reuse: Model-Based Reuse Characterization Schemes*, University of Maryland, Technical Report TR-2446, April 1990

[9]Basili, V. R., and H. D. Rombach, *Support for Comprehensive Reuse*, University of Maryland, Technical Report TR-2606, February 1991

[3]Basili, V. R., and R. W. Selby, Jr., "Calculation and Use of an Environment's Characteristic Software Metric Set," *Proceedings of the Eighth International Conference on Software Engineering*. New York: IEEE Computer Society Press, 1985

Basili, V. R., and R. W. Selby, Jr., *Comparing the Effectiveness of Software Testing Strategies*, University of Maryland, Technical Report TR-1501, May 1985

[3]Basili, V. R., and R. W. Selby, Jr., "Four Applications of a Software Data Collection and Analysis Methodology," *Proceedings of the NATO Advanced Study Institute*, August 1985

[5]Basili, V. R., and R. Selby, "Comparing the Effectiveness of Software Testing Strategies," *IEEE Transactions on Software Engineering*, December 1987

[9]Basili, V. R., and R. W. Selby, "Paradigms for Experimentation and Empirical Studies in Software Engineering," *Reliability Engineering and System Safety*, January 1991

[4]Basili, V. R., R. W. Selby, Jr., and D. H. Hutchens, "Experimentation in Software Engineering," *IEEE Transactions on Software Engineering*, July 1986

[2]Basili, V. R., R. W. Selby, and T. Phillips, "Metric Analysis and Data Validation Across FORTRAN Projects," *IEEE Transactions on Software Engineering*, November 1983

[2]Basili, V. R., and D. M. Weiss, *A Methodology for Collecting Valid Software Engineering Data*, University of Maryland, Technical Report TR-1235, December 1982

[3]Basili, V. R., and D. M. Weiss, "A Methodology for Collecting Valid Software Engineering Data," *IEEE Transactions on Software Engineering*, November 1984

[1]Basili, V. R., and M. V. Zelkowitz, "The Software Engineering Laboratory: Objectives," *Proceedings of the Fifteenth Annual Conference on Computer Personnel Research*, August 1977

Basili, V. R., and M. V. Zelkowitz, "Designing a Software Measurement Experiment," *Proceedings of the Software Life Cycle Management Workshop*, September 1977

[1]Basili, V. R., and M. V. Zelkowitz, "Operation of the Software Engineering Laboratory," *Proceedings of the Second Software Life Cycle Management Workshop*, August 1978

[1]Basili, V. R., and M. V. Zelkowitz, "Measuring Software Development Characteristics in the Local Environment," *Computers and Structures*, August 1978, vol. 10

Basili, V. R., and M. V. Zelkowitz, "Analyzing Medium Scale Software Development," *Proceedings of the Third International Conference on Software Engineering*. New York: IEEE Computer Society Press, 1978

[9]Booth, E. W., and M. E. Stark, "Designing Configurable Software: COMPASS Implementation Concepts," *Proceedings of Tri-Ada 1991*, October 1991

[9]Briand, L. C., V. R. Basili, and W. M. Thomas, *A Pattern Recognition Approach for Software Engineering Data Analysis*, University of Maryland, Technical Report TR-2672, May 1991

[5]Brophy, C. E., W. W. Agresti, and V. R. Basili, "Lessons Learned in Use of Ada-Oriented Design Methods," *Proceedings of the Joint Ada Conference*, March 1987

[6]Brophy, C. E., S. Godfrey, W. W. Agresti, and V. R. Basili, "Lessons Learned in the Implementation Phase of a Large Ada Project," *Proceedings of the Washington Ada Technical Conference*, March 1988

[2]Card, D. N., "Early Estimation of Resource Expenditures and Program Size," Computer Sciences Corporation, Technical Memorandum, June 1982

[2]Card, D. N., "Comparison of Regression Modeling Techniques for Resource Estimation," Computer Sciences Corporation, Technical Memorandum, November 1982

[3]Card, D. N., "A Software Technology Evaluation Program," *Annais do XVIII Congresso Nacional de Informatica*, October 1985

[5]Card, D. N., and W. W. Agresti, "Resolving the Software Science Anomaly," *The Journal of Systems and Software*, 1987

[6]Card, D. N., and W. W. Agresti, "Measuring Software Design Complexity," *The Journal of Systems and Software*, June 1988

[4]Card, D. N., V. E. Church, and W. W. Agresti, "An Empirical Study of Software Design Practices," *IEEE Transactions on Software Engineering*, February 1986

Card, D. N., V. E. Church, W. W. Agresti, and Q. L. Jordan, "A Software Engineering View of Flight Dynamics Analysis System," Parts I and II, Computer Sciences Corporation, Technical Memorandum, February 1984

Card, D. N., Q. L. Jordan, and V. E. Church, "Characteristics of FORTRAN Modules," Computer Sciences Corporation, Technical Memorandum, June 1984

[5]Card, D. N., F. E. McGarry, and G. T. Page, "Evaluating Software Engineering Technologies," *IEEE Transactions on Software Engineering*, July 1987

[3]Card, D. N., G. T. Page, and F. E. McGarry, "Criteria for Software Modularization," *Proceedings of the Eighth International Conference on Software Engineering*. New York: IEEE Computer Society Press, 1985

[1]Chen, E., and M. V. Zelkowitz, "Use of Cluster Analysis To Evaluate Software Engineering Methodologies," *Proceedings of the Fifth International Conference on Software Engineering*. New York: IEEE Computer Society Press, 1981

[4]Church, V. E., D. N. Card, W. W. Agresti, and Q. L. Jordan, "An Approach for Assessing Software Prototypes," *ACM Software Engineering Notes*, July 1986

[2]Doerflinger, C. W., and V. R. Basili, "Monitoring Software Development Through Dynamic Variables," *Proceedings of the Seventh International Computer Software and Applications Conference*. New York: IEEE Computer Society Press, 1983

Doubleday, D., *ASAP: An Ada Static Source Code Analyzer Program*, University of Maryland, Technical Report TR-1895, August 1987 (NOTE: 100 pages long)

[6]Godfrey, S., and C. Brophy, "Experiences in the Implementation of a Large Ada Project," *Proceedings of the 1988 Washington Ada Symposium*, June 1988

Hamilton, M., and S. Zeldin, *A Demonstration of AXES for NAVPAK*, Higher Order Software, Inc., TR-9, September 1977 (also designated SEL-77-005)

[5]Jeffery, D. R., and V. Basili, *Characterizing Resource Data: A Model for Logical Association of Software Data*, University of Maryland, Technical Report TR-1848, May 1987

[6]Jeffery, D. R., and V. R. Basili, "Validating the TAME Resource Data Model," *Proceedings of the Tenth International Conference on Software Engineering*, April 1988

[5]Mark, L., and H. D. Rombach, *A Meta Information Base for Software Engineering*, University of Maryland, Technical Report TR-1765, July 1987

[6]Mark, L., and H. D. Rombach, "Generating Customized Software Engineering Information Bases From Software Process and Product Specifications," *Proceedings of the 22nd Annual Hawaii International Conference on System Sciences*, January 1989

[5]McGarry, F. E., and W. W. Agresti, "Measuring Ada for Software Development in the Software Engineering Laboratory (SEL)," *Proceedings of the 21st Annual Hawaii International Conference on System Sciences*, January 1988

[7]McGarry, F., L. Esker, and K. Quimby, "Evolution of Ada Technology in a Production Software Environment," *Proceedings of the Sixth Washington Ada Symposium (WADAS)*, June 1989

[3]McGarry, F. E., J. Valett, and D. Hall, "Measuring the Impact of Computer Resource Quality on the Software Development Process and Product," *Proceedings of the Hawaiian International Conference on System Sciences*, January 1985

National Aeronautics and Space Administration (NASA), *NASA Software Research Technology Workshop* (Proceedings), March 1980

[3]Page, G., F. E. McGarry, and D. N. Card, "A Practical Experience With Independent Verification and Validation," *Proceedings of the Eighth International Computer Software and Applications Conference*, November 1984

[5]Ramsey, C. L., and V. R. Basili, *An Evaluation of Expert Systems for Software Engineering Management*, University of Maryland, Technical Report TR-1708, September 1986

[3]Ramsey, J., and V. R. Basili, "Analyzing the Test Process Using Structural Coverage," *Proceedings of the Eighth International Conference on Software Engineering*. New York: IEEE Computer Society Press, 1985

[5]Rombach, H. D., "A Controlled Experiment on the Impact of Software Structure on Maintainability," *IEEE Transactions on Software Engineering*, March 1987

[8]Rombach, H. D., "Design Measurement: Some Lessons Learned," *IEEE Software*, March 1990

[9]Rombach, H. D., "Software Reuse: A Key to the Maintenance Problem," *Butterworth Journal of Information and Software Technology*, January/February 1991

[6]Rombach, H. D., and V. R. Basili, "Quantitative Assessment of Maintenance: An Industrial Case Study," *Proceedings From the Conference on Software Maintenance*, September 1987

[6]Rombach, H. D., and L. Mark, "Software Process and Product Specifications: A Basis for Generating Customized SE Information Bases," *Proceedings of the 22nd Annual Hawaii International Conference on System Sciences*, January 1989

[7]Rombach, H. D., and B. T. Ulery, *Establishing a Measurement Based Maintenance Improvement Program: Lessons Learned in the SEL*, University of Maryland, Technical Report TR-2252, May 1989

[6]Seidewitz, E., "Object-Oriented Programming in Smalltalk and Ada," *Proceedings of the 1987 Conference on Object-Oriented Programming Systems, Languages, and Applications*, October 1987

[5]Seidewitz, E., "General Object-Oriented Software Development: Background and Experience," *Proceedings of the 21st Hawaii International Conference on System Sciences*, January 1988

BI-11

[6]Seidewitz, E., "General Object-Oriented Software Development with Ada: A Life Cycle Approach," *Proceedings of the CASE Technology Conference*, April 1988

[9]Seidewitz, E., "Object-Oriented Programming Through Type Extension in Ada 9X," *Ada Letters*, March/April 1991

[4]Seidewitz, E., and M. Stark, "Towards a General Object-Oriented Software Development Methodology," *Proceedings of the First International Symposium on Ada for the NASA Space Station*, June 1986

[9]Seidewitz, E., and M. Stark, "An Object-Oriented Approach to Parameterized Software in Ada," *Proceedings of the Eighth Washington Ada Symposium*, June 1991

[8]Stark, M., "On Designing Parametrized Systems Using Ada," *Proceedings of the Seventh Washington Ada Symposium*, June 1990

[7]Stark, M. E. and E. W. Booth, "Using Ada to Maximize Verbatim Software Reuse," *Proceedings of TRI-Ada 1989*, October 1989

[5]Stark, M., and E. Seidewitz, "Towards a General Object-Oriented Ada Lifecycle," *Proceedings of the Joint Ada Conference*, March 1987

[8]Straub, P. A., and M. V. Zelkowitz, "PUC: A Functional Specification Language for Ada," *Proceedings of the Tenth International Conference of the Chilean Computer Science Society*, July 1990

[7]Sunazuka, T., and V. R. Basili, *Integrating Automated Support for a Software Management Cycle Into the TAME System*, University of Maryland, Technical Report TR-2289, July 1989

Turner, C., and G. Caron, *A Comparison of RADC and NASA/SEL Software Development Data*, Data and Analysis Center for Software, Special Publication, May 1981

Turner, C., G. Caron, and G. Brement, *NASA/SEL Data Compendium*, Data and Analysis Center for Software, Special Publication, April 1981

[5]Valett, J. D., and F. E. McGarry, "A Summary of Software Measurement Experiences in the Software Engineering Laboratory," *Proceedings of the 21st Annual Hawaii International Conference on System Sciences*, January 1988

[3]Weiss, D. M., and V. R. Basili, "Evaluating Software Development by Analysis of Changes: Some Data From the Software Engineering Laboratory," *IEEE Transactions on Software Engineering*, February 1985

[5]Wu, L., V. R. Basili, and K. Reed, "A Structure Coverage Tool for Ada Software Systems," *Proceedings of the Joint Ada Conference*, March 1987

[1]Zelkowitz, M. V., "Resource Estimation for Medium-Scale Software Projects," *Proceedings of the Twelfth Conference on the Interface of Statistics and Computer Science*. New York: IEEE Computer Society Press, 1979

[2]Zelkowitz, M. V., "Data Collection and Evaluation for Experimental Computer Science Research," *Empirical Foundations for Computer and Information Science* (Proceedings), November 1982

[6]Zelkowitz, M. V., "The Effectiveness of Software Prototyping: A Case Study," *Proceedings of the 26th Annual Technical Symposium of the Washington, D. C., Chapter of the ACM*, June 1987

[6]Zelkowitz, M. V., "Resource Utilization During Software Development," *Journal of Systems and Software*, 1988

[8]Zelkowitz, M. V., "Evolution Towards Specifications Environment: Experiences With Syntax Editors," *Information and Software Technology*, April 1990

Zelkowitz, M. V., and V. R. Basili, "Operational Aspects of a Software Measurement Facility," *Proceedings of the Software Life Cycle Management Workshop*, September 1977

NOTES:

[1]This article also appears in SEL-82-004, *Collected Software Engineering Papers: Volume I*, July 1982.

[2]This article also appears in SEL-83-003, *Collected Software Engineering Papers: Volume II*, November 1983.

[3]This article also appears in SEL-85-003, *Collected Software Engineering Papers: Volume III*, November 1985.

[4]This article also appears in SEL-86-004, *Collected Software Engineering Papers: Volume IV*, November 1986.

[5]This article also appears in SEL-87-009, *Collected Software Engineering Papers: Volume V*, November 1987.

[6]This article also appears in SEL-88-002, *Collected Software Engineering Papers: Volume VI*, November 1988.

[7]This article also appears in SEL-89-006, *Collected Software Engineering Papers: Volume VII*, November 1989.

[8]This article also appears in SEL-90-005, *Collected Software Engineering Papers: Volume VIII*, November 1990.

[9]This article also appears in SEL-91-005, *Collected Software Engineering Papers: Volume IX*, November 1991.